Is the cloud ready for low-latency messaging?

Shaun Laurens
Head of Engineering USA - Adaptive

Is the cloud ready for low-latency messaging?

- It depends...

Let’s take a step back.

- Cloud and Finance have taken different paths
- In finance: last 15 years spent on more predictable latency
- We have had some shared lessons, for example how to measure performance
- Latency distributions offer much more insight than averages

Cloud

- AWS kick started around 20 years ago, with Toys R Us
- They faced very different problems to finance
- Their optimization goals led to API driven services, independent teams, 24x7, partial failures
- TCP dominates, and is very different UDP

Key Resources

- Servers
- Disks
- Networks

Servers

- Back in the day, gaming was a big influence on CPUs
- Cloud is a driver today, again with specific optimization goals such as power consumption & density
- Newest CPUs are often available first on cloud
- Upgrades are typically a simple, quick process on cloud

Servers and system architecture

- We're seeing more systems getting built out of many smaller services deployed on fewer machines
- On Linux, loopback is fast; shared memory is faster
- Leverage shared memory for low latency interprocess messaging

Interprocess Messaging

- These kind of optimizations are available on cloud assuming that you have full allocation of at least a socket, and polite neighbors

Back to Science

- On the cloud, we can’t just call our server vendor and get performance tuning guidelines
- Instead, we need to revert back to the scientific method
- Build a theoretical model of your application, benchmark, test, run experiments, and work toward lower and more predictable latency

Networks

- Unlike cloud, finance has long favored UDP multicast
- TCP brings with it congestion and flow control, and latency
- TCP's slow start after idle is especially problematic for latency sensitive financial applications

UDP on Cloud

- Unlike on-prem, the cloud is not over provisioned & private
- The cloud providers require you to be a good citizen
- Being a good citizen brings us back to congestion and flow control

UDP on Cloud

- Things have been getting better on the wider internet with UDP, for example with QUIC as used by Facebook, Google and others
- QUIC does give more predictable latency
- Multicast - as we often use - remains a problem

Multicast on Cloud

- Some cloud providers claim to support multicast today
- Latency is neither low nor predictable
- Talk of real multicast arriving at some stage
- Can be simulated, to an extent, with overlay networks
- Existing multicast players are unlikely to be fit for cloud with a focus on rate limiting

Storage

- We often need to be able to store and replay our communications
- Storing data on the cloud can be a bit counterintuitive
- Local NVME storage is typically ephemeral, yet remote storage offers challenging tail latencies
- Use pure in-memory storage for the best latency profile, but this is not HA. If you need to store more, use remote disk

Syscalls

- Profiling highlights the high overhead of syscalls on the cloud
- We have to move towards batching to reduce this overhead
- To gain the most, your applications need to be built like this from the ground up
- Going beyond batching, we can adopt kernel bypass

Kernel Bypass on the Cloud

+ DPDK brings Kernel bypass to several cloud providers
+ Linux features are also moving in the right direction, with io-uring in the latest kernels

So, what can we achieve?

- For the last 10 years, Adaptive & Real-Logic have been working on Aeron
- Aeron builds upon the past 30 years of networking
- Aeron Transport is a UDP (or IPC) based messaging transport with flow control and congestion control built in
- Aeron batches throughout the stack, and we’re able to plug in kernel bypass (including DPDK)

Side note: Distributed Consensus

- Distributed consensus, plus replicated state machines, provide a fault tolerant container suitable for the cloud
- Paxos, Viewstamped Replication & RAFT are typical implementations. Virtual Synchrony is somewhat related
- Historically, distributed consensus has been too slow, with typical implementations suffering latency & throughput limitations

So, what can we achieve?

- For systems which need storage, and the ability to replay messaging later, we offer Aeron Cluster.
- Aeron Cluster allows to replicate state in real-time across multiple nodes, allowing systems to run ‘Cattle’ style
- We replicate, on cloud, at a rate in the millions of messages per second, with around 100μs latency (RTT). That drops to 20μs on prem.

Thank you!